REMARKS

Claims 1-19 are pending.

The claims stand rejected under 35 U.S.C. 103(a) as obvious in view of Patel, Johnson, and, for some claims, Peled. Applicants traverse the rejections.

To establish *prima facie* obviousness, there must be some suggestion or motivation to modify the reference or combine the reference teachings. There is no such suggestion or motivation here.

Patel discloses the improvement of fetch rates in trace caches by employing branch promotion and trace packing. Branch promotion removes the overhead resulting from dynamic branch prediction by applying static branch prediction to strongly biased branches. See, e.g., Patel, Abstract. Trace packing packs as many instructions as possible into a pending trace so that more instructions segments may be fetched during a single fetch cycle. See, e.g., Patel, Abstract. Patel neither teaches nor suggests that the fetch rates may be improved by reversing the order of the instructions in the traces. Moreover, there is would be no motivation to do so, since reversing instruction order would not appear to improve fetch rates. Hence, there would be no motivation to modify Patel with Johnson to arrive at the embodiments of the present invention.

The Office Action asserts that it would be obvious to modify the instruction segment of Patel with the teaching of Johnson in order to store instructions of an instruction trace in reverse order "so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved." See, e.g., Office Action, page 3, item 8. The assertions have been carefully analyzed and found to be erroneous for at least the following reasons.

By definition, a trace is a sequence of dynamically executed instructions, which may originally reside in non-continuous portions of the program memory, starting with a single entry instruction and ending with multiple exit instructions. For a typical trace, the head of the trace,

i.e., the first instruction in a sequence, is followed by the next executable instruction in the sequence, then the next, and so on.

If the Office Action's assertions are to be believed, the first instruction is accessed and modified more than the second, third, etc., instructions. However, such is contrary to the known operation of the typical trace.

It is unclear how accessing the first instruction in a trace more frequently than the second instruction, and accessing the second instruction more frequently than the third, and so on, would improve the performance of a trace. Since the trace defines sequential instructions, which perform a particular operation, accessing the instructions in decreasing frequency would in no way advance the completion of the particular operation. Indeed, such access of the trace would hinder the completion, thereby defeating the purpose of the trace.

Moreover, it is unclear how modifying the first instruction in a trace more frequently than the second instruction, and modifying the second instruction more frequently than the third, and so on, would improve the performance of a trace. Again, since the trace defines sequential instructions, which perform a particular operation, modifying the instructions in decreasing frequency would in no way advance the completion of the particular operation. Indeed, such modification would result in a different operation, thereby, defeating the purpose of the trace.

Therefore, the Office Action's asserted motivation for modifying Patel with Johnson makes little sense.

Furthermore, even if the head of the trace could be more frequently accessed and modified, the Office Action has provided no explanation of how such would improve the fetch rates of the trace cache of Patel.

Indeed, there is no motivation to reverse the instructions in a Patel trace. As stated previously, Patel discloses using branch promotion to improve cache fetch rates. The purpose

of branch promotion is to reduce the dynamic branching of strongly biased traces by applying static branches (or predictions). It is unclear how reversing the instructions so that the first instruction is listed last in the trace improves the branch promotion technique. Reversing the instructions would not appear to reduce the dynamic branching of strongly biased traces. Moreover, such would not appear to improve the fetch rates. As such, a person of ordinary skill in the art would not be motivated to reverse the instructions in a trace, while using branch promotion, to improve fetch rates.

Patel also discloses using trace packing to improve cache fetch rates. The purpose of trace packing is to increase the number of instructions fetched per fetch cycle. It is unclear how reversing the instructions so that the first instruction is listed last in the trace improves the trace packing technique. A reversal would not appear to increase trace packing capabilities. Moreover, such would not appear to improve the fetch rates. As such, a person of ordinary skill in the art would not be motivated to reverse the instructions in a trace, while using trace packing, to improve fetch rates.

Accordingly, the Office Action has failed to establish a *prima facie* case of obviousness over Patel in view of Johnson.

The deficiencies are not corrected by Peled. Since Peled fails to provide any motivation for modifying Patel with Johnson, there is no motivation, therefore, to modify Patel with Johnson and Peled to arrive at the claimed invention. Accordingly, the Office Action has failed to establish a *prima facie* case of obviousness over Patel in view of Johnson in further view of Peled.

For at least these reasons, the claims are believed to be patentable over the cited references, individually and in combination. Withdrawal of the rejections is, therefore, respectfully requested.

CONCLUSION

The claims are believed to be allowable. Hence, it is requested that the amendment be entered.

The Office is hereby authorized to charge any additional fees under 37 C.F.R. §1.16 or §1.17 or credit any overpayment to Deposit Account No. 11-0600.

Should the Examiner have any questions concerning this matter, he is invited to contact Applicants' undersigned attorney at 202.220.4200.

Respectfully submitted,

Date: Feb. 28, 2005

Cassandra T. Swain, Ph.D.
Registration No. 48,361
(Attorney for Intel Corporation)

KENYON & KENYON
1500 K Street, N.W.
Washington, D.C. 20005
Ph.: (202) 220-4200
Fax.: (202) 220-4201